DOCUMENT RESUME

ED 247 886 IR 011 235

AUTHOR Vincent, A. T.
TITLE Computer-Assisted Support for Blind Students: The Use of a Microcomputer Linked Voice Synthesizer. CAL Research Group Technical Report No. 10.
INSTITUTION Oper. Univ., Walton, Bletchley, Bucks (England).
PUB DATE Jan 81
NOTE 21p.; Paper presented at CAL 81 Symposium on Computer Assisted Learning (Leeds, England, April 8-10, 1981).
AVAILABLE FROM Institute of Educational Technology, The Open University, Walton Hall, Milton Keynes, MK7 6AA, England.
PUB TYPE Viewpoints (120) -- Reports - Descriptive (141)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Accessibility (for Disabled); *Artificial Speech; *Blindness; *Computer Software; Foreign Countries; Higher Education; *Microcomputers; Programing; Systems Development
IDENTIFIERS Open University (Great Britain); *Voice Synthesizers

ABSTRACT

Designed primarily to enable blind students to more easily use the Open University's Student Computer Services, a project was implemented to generate computer programs (in BASIC) for a limited configuration microcomputer with synthetic speech as the only output program. Three techniques can be identified with synthetic speech generation: speech-by-rule, look-up tables, and compressed digital speech. To provide a programmer support system that would facilitate the incorporation of synthetic speech output into computer programs, a dictionary/conversion program was developed which replaces PRINT, INPUT, DATA, and string statements in a program written for video display unit output, provides the phonetic commands for a voice synthesizer, and incorporates subroutines to assist blind students in using the program. Essential program subroutines that have been identified include synthetic speech output, repetition of speech output, validation and changing of input data, numeric phonemes, and output of errors. The lack of pitch or speed control limits the speech quality of the voice synthesizer used (which runs on a TRS-80 microcomputer), but its ease of use and relatively low cost compensate for the limited quality. Six references are listed. (LMM)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED247886

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
This document has been reproduced as received from the person or organization originating it.
Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

Computer Assisted Learning Research Group

Computer-assisted support for blind students - a microcomputer linked Voice Synthesiser

A.T.Vincent

CAL Research Group Technical Report No.10

IR011235

Walton Hall,
Milton Keynes,
England MK7 6AA

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
E.S. Henderson
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Computer-assisted support for blind students

- the use of a microcomputer linked Voice Synthesizer

.....

Paper submitted to CAL 81

A.T. Vincent
The Open University

January 1981

IR01123'5'

Abstract

Computer-assisted support for blind students - the use of a microcomputer linked Voice Synthesizer.

A.T. Vincent
The Open University, 70 Manchester Road,
Chorlton-cum-Hardy, Manchester M21 1PQ.

The generation of computer programs (in BASIC) for a limited configuration microcomputer with synthetic speech as the only output medium is described. This has involved the development of a dictionary/conversion program that replaces PRINT, INPUT, DATA and string statements in a program written for VDU output, provides the phonetic commands for a Voice Synthesizer, and incorporates subroutines that will assist a blind student to use the programs.

.....

Paper presented at CAL 81

CAL Research Group Technical Report No.10

April 1981

Computer-assisted support for blind students - the use of a microcomputer linked Voice Synthesizer.

A.T. Vincent
The Open University, 70 Manchester Road,
Chorlton-cum-Hardy, Manchester M21 1PQ.

Developments in microcomputer technology have created new opportunities for the application of computers in teaching. This is particularly true where cost has previously been an inhibiting factor; a significant feature in providing computer-assisted support to blind students. This paper describes a project that commenced in 1980 with the objective of producing computer programs (written in BASIC) that could be used by blind students using a low-cost microcomputer system with synthetic speech as the only output medium.

Blind students in the Open University

Since 1970, the Open University has made extensive use of computers to support the distance teaching of undergraduate and associate students[1]. Currently, over 30,000 students each year are advised or required to make use of a computer facility either in a local Study Centre or at Summer School. Generally, this involves access via a teletype to one of three main-frame computers. Various microcomputers are also used at Summer Schools. The computers are used for a variety of activities which include programming, data-handling, simulation and tutorials.

For the blind student, there are two methods which can be adopted to make use of the Student Computer Service facilities offered by the Open University. Firstly, a normal sighted person can be present at the terminal to relay details of computer generated questions and responses to the student. This arrangement can be satisfactory,

however, it does require the presence of a second person for long periods, and it excludes independent use. The second method allows independent access by using the commercially available Optacon [2]. This device transduces information that appears under a lens into a character representation on a pad of vibrating pins. Hence, computer output on a screen or teletype can be scanned and read. At present, the cost of this device has resulted in only a small number of students using this method of access. Generally, blind students avoid courses where there is a significant amount of computer use involved. The number of students taking programming courses is very small.

Synthetic Speech

Direct synthetic speech output from a computer terminal would help to overcome the problems experienced by blind students using a VDU or teletype. Some progress has been made in the U.S.A. with the development of commercially available talking terminals [3]; however, the wider application of these new peripherals has probably been inhibited by their cost.

Good quality synthetic speech is difficult to achieve because of the dynamic properties of the human-speech mechanism that have to be simulated. It is well known what essential components are required to generate human speech [4]; research continues into the development of hardware and software to exploit this knowledge. Synthetic speech, however, is still not acceptable to critical ears but may well be acceptable if this means of communication creates opportunities that would not otherwise exist.

Three techniques can be identified with the generation of synthetic speech: speech-by-rule, look-up tables, and compressed digital speech.

Speech-by-rule

Computer-generated speech from unrestricted text take place in two discrete steps. Firstly, the text is converted into a series of commands that are associated with a voice generation unit. Secondly, the commands are interpreted by the voice synthesizer to produce audible output. Commonly, this is achieved by using a speech-by-rule computer program which translates text into its phonetic equivalents using a restricted set of commands that are available to the voice synthesizer which, in turn, creates synthetic speech from the phonetic commands.

A comprehensive speech-by-rule algorithm has been reported[5] which uses less than 1000 pronunciation rules to produce generally acceptable speech quality. The algorithm was originally written for a PDP11/45 - the rules occupying approximately 11K of memory, pronunciation code and tables 5K, and a further 2K for the interactive display and maintenance of tables. The algorithm has been implemented subsequently on a microcomputer system (Intel 8080A-based) at the U.S. National Institute of Health[6].

Speech-by-rule synthesis has a particular advantage in that it has an unlimited vocabulary.

Look-up tables

This technique differs from speech-by-rule in step 1; in this case the phonetic equivalent of words are derived from a table of information rather than by synthesis. This requires alternative methods for generating the phonetic equivalents and storing the information in memory prior to extraction and transmission to the voice synthesizer when a word or phrase is output.

The vocabulary is limited by the size of the look-up table which, in turn, is directly dependent on the amount of memory available. Typically, 100 words and their phonetic equivalents can be stored in a table occupying 2K of memory. An additional amount of memory is required to store, retrieve and transmit the phonetic commands to the voice synthesizer.

Compressed digital speech

At the simplest level, speech can be recorded in an analogue form on magnetic tape. The addition of addressing and timing information on a second track allows the recorded audio to be replayed under computer control. The serious disadvantage of this system is the slow serial access that a magnetic tape provides.

The access time can be considerably improved by using a random-access audio disk; this has been successfully implemented for the Plato computer-based education system. Twenty minutes of high quality speech can be pre-recorded with the advantage of random access and minimal data transfer for message selection. However, a sophisticated disk drive is required to manage the disk.

Open University project

The prime objectives of the project described in this paper are (a) to provide synthetic speech output for blind students using computer programs (BASIC) on a microcomputer, and (b) to use a low-cost commercially available speech synthesis system that could eventually lead to it being made widely available to students.

The first objective excludes the need to consider the support for computer programming. This simplifies the project as, on the whole,

an individual computer program has a limited text output requirement which can be met by a restricted vocabulary that incorporates individual alpha-numeric and other keyboard characters.

The second objective proved to be very restrictive - the system finally selected was a Tandy TRS-80 microcomputer and Voice Synthesizer.

Hardware

Two configurations were chosen: one for program development; the other for use by blind students to run programs.

Development system	Student system
keyboard/VDU	keyboard
48K CPU	16K CPU
2 disk drives	cassette recorder
cassette recorder	voice synthesizer
line printer	
voice synthesizer	

It was decided not to include a VDU with the student system during the development stage as experience could be gained of program requirements by a wide range of users (not necessarily blind students) when visual output was not available. In practice, a monitor or TV output can be provided readily which will be useful if a blind student is assisted by, for example, a tutor.

Voice Synthesizer

The TRS-80 Voice Synthesizer is capable of producing 62 phonemes; the commands to produce these speech units are generated from the microcomputer software.

As the Voice Synthesizer is driven by phonetic commands, it can be used as the second stage for both synthesis-by-rule and look-up table techniques. The choice between these techniques was determined by the nature of the computer programs that would be used by students. Synthesis-by-rule was eliminated because (a) an individual program has a restricted output requirement that can be met by an appropriate vocabulary and (b) considerable time would have been needed to develop a software package for this technique.

The interface of the Voice Synthesizer has been designed such that a BASIC PRINT can be used as an output command. This has been achieved by address mapping the interface into the last 32 locations (992-1023) of the video display. Any method of PRINTing, POKEing or loading into these locations, combined with a 'device select' character (a "?"), causes the Voice Synthesizer to produce the specified sounds. A second "?" closes the device.

The following BASIC program illustrates one method of inputting any combination of the Voice Synthesizer's 62 phonetic commands and producing the equivalent sound:

```
10 INPUT"ENTER PHONEME(S)";A$
20 PRINT@992,"? ";A$;" ?";
```

which can be compared with the input/output statements of text for a VDU:

```
10 INPUT"ENTER WORD";A$
20 PRINT@992,A$
```

This relatively easy adaptation of a program to provide synthetic speech output is an important feature in relation to the development of programs to use this facility.

Use of phonemes

Virtually any word in the English language can be produced using the 62 phonemes provided in the TRS-80 Voice Synthesizer. In the absence of a translator, a dictionary of words and equivalent phonetic commands is created within a look-up table. As an aid to this, the sound units are assigned special symbols. For example, the vowel sound in "beg" is represented by EH1; the shorter version of this sound is EH3 and is used in a word like "jacket". The number at the end of the symbol indicating the duration of the sound (an inverse relationship).

An example of this construction is:

word	-	seven
phoneme symbols (for notation and reference)	-	S EH3 EH2 V EH2 N N
ASCII symbols (to be PRINTed)	-	S 5 4 V 4 N N

The BASIC statement which would result in the Voice Synthesizer "speaking" the word "seven" would be:

```
10 PRINT@992,"? S54V4NN ?"
```

Programs incorporating synthetic speech output

Initially, trial programs were written for the TRS-80 microcomputer that incorporated output to the Voice Synthesizer with phonetic commands extracted from a dictionary, developed manually outside of the microcomputer, to assist in the production of the appropriate PRINT

statements. This method did not prove to be entirely satisfactory, and the need for a dictionary resident within the microcomputer soon became apparent. In addition, facilities to experiment with phonetic constructions, and to provide updating, printing and sorting for the dictionary were also identified.

The development of a memory resident, interactive dictionary proved to be a valuable aid to the programmer; however, there still remained a time-consuming step when programs were being revised or debugged. Running a program where all the output is via the Voice Synthesizer cannot be accelerated in the same way as with video display where PRINT statement output can be presented at a rate that is only limited by the output rate to the VDU. Normal speech rate is much slower than this, and cannot be speeded-up as it would then become undistinguishable.

To overcome this difficulty, it has proved helpful to have two versions of any program - one with video output and the other with synthetic speech output. This has resulted in a major revision of the dictionary support software to include the automatic conversion of a source program (video output) to an object program (synthetic speech output) with the provision to (a) convert all text in strings to phonetic commands extracted from the dictionary, (b) rewrite all PRINT, INPUT, DATA and string statements, and (c) add subroutines to aid the subsequent use of the object program.

The method adopted is illustrated in Figure 1. The dictionary/conversion program is first loaded from disk into the development microcomputer system where, after initialisation, the facility exists to update or extend the dictionary using the following menu:

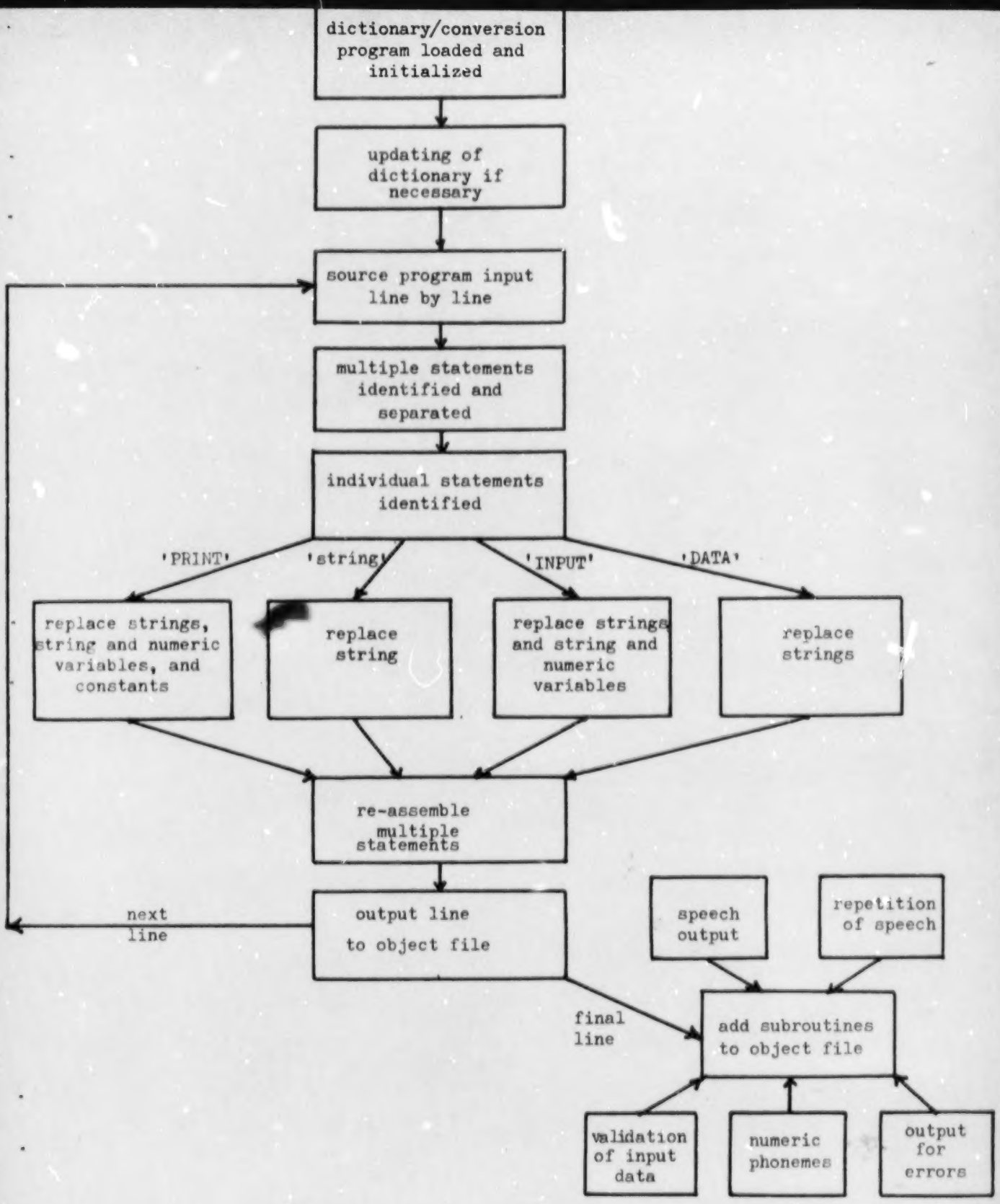

Figure 1. Conversion of course (video output) to object program (speech output).

System Command < >	Action
<.>	ENDS THE PROGRAM
</>	UPDATES THE DICTIONARY DISK FILE
<;>	LISTS PART OF THE DICTIONARY ON THE VDU
<#>	PRINTS DICTIONARY ON LINE PRINTER
<WORD>	REQUESTS PHONETIC COMMANDS IF WORD IS NOT IN DICTIONARY; OTHERWISE, RETURNS COMMANDS TO VDU AND VOICE SYNTHESIZER
<WORD> <@>	CHANGES PHONETIC COMMANDS FOR SPECIFIED WORD
<$>	OBJECT/SOURCE PROGRAM CONVERSION

The final system command initiates the conversion of a pre-defined source program held on disk. The program is input line-by-line. Lines are separated into individual statements (where necessary) before the PRINT, INPUT, DATA and string statements are converted to synthetic speech program formats. If strings within these statements are found to have words that have not already been recorded in the dictionary, this facility is re-called to allow experimentation with phonetic command combinations until a satisfactory sound is obtained. The phonetic combination is then entered into the dictionary before returning to the object program to continue the conversion.

A typical line conversion is:

source line:

```
10 PRINT"COMPUTER SCIENCE"
```

object line:

```
10 VZS="K7MPY(UQ/ SA;*#NTS":GOSUB 50000
```

In addition to the word/phonetic command conversion of the string, the line is restructured to meet the needs of a synthetic speech output subroutine (see (a) below) that is added to the object program.

After all of the course program lines have been converted, several subroutines are added to facilitate the final use of the program by a blind student on the limited configuration microcomputer previously described. To date the following subroutines have been identified as essential:

(a) synthetic speech output: an alternative method to that using PRINT@992,"? phonetic commands ?" has been developed to accommodate lengthy strings. The PRINT method is restricted to a maximum consecutive output of 32 characters by the capacity of the Voice Synthesizer buffer. The speech output subroutine handles strings longer than 32 characters by the introduction of short delay loops at regular intervals, e.g. after every word. This has no effect on the rate of speech output but does allow the buffer time to clear.

(b) repetition of speech output: this subroutine allows a student to request a repeat of the last output from the Voice Synthesizer. It is activated by pressing the 'space-bar' (as an alternative to the 'enter' key for continuing) at appropriate times.

(c) validation and changing of input data: it is essential that during the entry of data to an INPUT statement that the student responses can be echoed via the Voice Synthesizer, and the opportunity given to change the data. This subroutine is designed for this function. If data is satisfactory the 'enter' key is pressed, if not, new data is typed. The first character received indicates that a change is being made - this method limits the number of keyboard entries, i.e. a 'change data' command is not necessary.

(d) numeric phonemes (Ø to 9): this subroutine is provided to allow numeric values to be output via the Voice Synthesizer. Numbers are 'spoken' character-by-character including '+', '-', '.' and the exponent.

(e) output of errors: if a system error arises during the course of running the object program, this subroutine provides for a suitable error message to be output via the Voice Synthesizer.

After the addition of the subroutines, the object program is stored on disk ready to use. It can be accessed by the programmer for testing - if it is not satisfactory, it is possible to return to the source program for debugging or revision, followed by a further pass through the dictionary/conversion program to produce a revised object program. Small changes to the object program can be made directly by the normal editing facilties available on the TRS-80.

The final version of the object program is copied to cassette magnetic tape for use by a student on the limited configuration microcomputer.

Initial trials

During the first phase of the project described in this paper, two existing programs for video display output have been used for the developmental testing of the dictionary/conversion program. In addition, a special program has been written for use by a student to gain familiarity with the microcomputer keyboard. These programs are:

(a) keyboard awareness - designed to allow a blind student 'explore' the keyboard by (i) typing any key, (ii) responding to random requests to type particular keys, and (iii) responding to requests

to type selected words. In each case the microcomputer responds through the Voice Synthesizer with characters, words or messages.

(b) statistical analysis - data-handling is an important application in the project. This particular program inputs and validates data, fits the function $y = mx + c$, calculates standard deviations, and outputs the results to the Voice Synthesizer. An important feature of the program is the validation of the data which would be common to many data-handling routines.

(c) conditional registration - an interactive program that assists with the completion of a Conditional Registration (course selection) form. Questions are asked relative to the various sections on the form and responses checked - finally, the responses are output to the cassette magnetic tape for the subsequent re-creation of a hardcopy of the completed form.

Preliminary findings

The first phase of the project described in this paper has concentrated on (a) the production of a programmer support system to facilitate the incorporation of synthetic speech output into computer programs, and (b) the identification of supplementary subroutines that will assist a blind student to use a computer program.

The programmer support system has been provided by the dictionary/conversion software already described; this has proved to be an essential facility. The availability of a source program that has VDU output and a matching object program with synthetic speech output (provided by the dictionary/conversion program) is a key factor at the program development stage.

Seven subroutines have so far been identified as being necessary to incorporate into the final version of a program to be used on a limited configuration microcomputer with only synthetic speech output. The subroutines can readily be added using the dictionary/conversion program. If further subroutines are found to be necessary during future trials, it would only require a modification of the dictionary/conversion program followed by the re-conversion of the source programs.

The speech quality of the Voice Synthesizer used for the project is limited by the lack of pitch or speed control. This is evident in the monotone nature of the speech; however, this has not proved to be a problem. The programs developed so far are interactive with only short periods of synthetic speech at any one time. Any lack of quality is compensated for by the ease of use and relatively low cost of the Voice Synthesizer.

With recent announcements of new higher quality/lower cost synthetic speech chips, the prospects for blind students gaining wider access to computers is very encouraging.

The assistance of G. Stephenson in the development of the dictionary software, and P. Stableford in the phonetic translations is gratefully acknowledged.

References

1. Bramer M.A., CAL Research Group Technical Report, No.1, The Open University, Milton Keynes, England, 1980.

2. Optacon®, Telesensory Systems, Inc., Palo Alto, California, U.S.A.

3. Maryland Computer Services, Inc., Maryland, U.S.A.
Kurzweil R., The Kurzweil Report, Kurzweil Computer Products, Cambridge Mass., U.S.A., 1979.

4. Flanagan J.L., Coker C.H., Rabiner L.R., Schafer R.W. and Umeda N., IEEE Spectrum, Vol.7, No.10, October 1970.

5. McIlroy M.D., Computer Science Technical Report, No.14, Bell Telephone Laboratories, Murray Hill, N.Y., U.S.A., March 1974.

6. Songe D.C., Allen S.I., Plexico P.S. and Morford R.A., IEEE Spectrum, Vol.17, No.5, May 1980.

COMPUTER-ASSISTED LEARNING RESEARCH GROUP

THE OPEN UNIVERSITY

List of Technical Reports

These reports may be obtained (free of charge but subject to availability) from:-

Ann Jones,
Institute of Educational Technology,
The Open University,
Walton Hall,
MILTON KEYNES. MK 7 6AA
England.

Report no.	Title and Author
1	M.A. Bramer (1980) Using Computers in Distance Education: The first ten years of the British Open University.
2	P.G. Butcher (1979) Computer Assisted Learning in Chemistry at the Open University.
3	M.A. Bramer (1980) COMAL-80 - Adding Structure to BASIC.
4	A. Jones (1980) An evaluation of the SOLO programming language: A pilot study of its use at summer school.
5	A. Jones, T. O'Shea (1981) An evaluation of tutorial CAL at the Open University: the use of Cicero in SDT286.
6	E.L. Whitelegg (1981) An evaluation of tutorial CAL at the Open University: the use of MERLIN in ST291.
7	M.W. Lewis (1980) Improving SOLO's User-Interface: An Empirical study of user behaviour and a proposal for cost-effective enhancements to SOLO.
8	A.T. Vincent (1980) Some applications of a microcomputer.
9	T. O'Shea, A. Floyd (1981) Recording Children's mathematical behaviour.
10	A.T. Vincent (1981) Computer-assisted support for blind students: the use of a microcomputer-linked voice synthesizer.
11	P. Bacsich (1981) Viewdata-style delivery mechanisms for CAL.
12	P.J. Murphy (1981) Biological simulations in distance learning.
13	P.J. Murphy (1981) Computer simulation in Education: Genetics and Evolution.
14	I. Every (1981) Graphics and animation in teaching dialogues.
15	P.J. Murphy, E. Scanlon, E.L. Whitelegg (1981) An evaluation of simulation CAL in the Science Faculty during 1980.
16	M. Eisenstadt, J. Laubsch, J.H. Kahney (1981) Creating pleasant programming environments for cognitive science students.
17	J. Laubsch, M. Eisenstadt (1981) Domain specific debugging aids for novice progreammers.
18	J.Roberts (1981) Are fuzzy grids really necessary?

BEST COPY AVAILABLE